

# **An Agent-Based Simulation Model for Organizational Analysis**

## **Student Submission** **(Modeling and Simulation)**

**Sui Ruan<sup>1,\*</sup>**

**E-mail: [sruan@engr.uconn.edu](mailto:sruan@engr.uconn.edu)**

**University of Connecticut, Dept. of Electrical and Computer Engineering**

**Swapna S. Gokhale<sup>1</sup>**

**E-mail: [ssg@engr.uconn.edu](mailto:ssg@engr.uconn.edu)**

**University of Connecticut, Dept. of Computer Science and Engineering**

**Krishna R. Pattipati<sup>1,2</sup>**

**<sup>1</sup>University of Connecticut, Dept. of Electrical and Computer Engineering**

**371 Fairfield Road, Unit 1157**

**Storrs, CT 06269-1157**

**Fax: 860-486-5585**

**Phone: 860-486-2890**

**E-mail: [krishna@engr.uconn.edu](mailto:krishna@engr.uconn.edu)**

<sup>1</sup>This work has been supported by the Office of Naval Research under Contract N00014-00-1-010

<sup>2</sup>Correspondence: [krishna@engr.uconn.edu](mailto:krishna@engr.uconn.edu)

\*Student

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE <b>JUN 2006</b>		2. REPORT TYPE		3. DATES COVERED <b>00-00-2006 to 00-00-2006</b>	
4. TITLE AND SUBTITLE <b>An Agent-Based Simulation Model for Organizational Analysis</b>				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) <b>University of Connecticut, Department of Electrical and Computer Engineering, 371 Fairfield Road Unit 1157, Storrs, CT, 06269-1157</b>				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT <b>Approved for public release; distribution unlimited</b>					
13. SUPPLEMENTARY NOTES <b>The original document contains color images.</b>					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES <b>35</b>	19a. NAME OF RESPONSIBLE PERSON
a. REPORT <b>unclassified</b>	b. ABSTRACT <b>unclassified</b>	c. THIS PAGE <b>unclassified</b>			

# **An Agent-Based Simulation Model for Organizational Analysis**

Sui Ruan, Swapna S. Gokhale, and Krishna R. Pattipati

## **Abstract**

In many fields, including engineering, management, and organizational science, simulation-based on computational organization theory has been used to gain insight into the degree of match (“congruence”) between the organization (people, work processes and structure) and the tasks carried out by the organization. Simulation helps identify the bottlenecks, and improve the quality and efficiency of an organization.

In this paper, we propose an approach based on the congruence model for analyzing and simulating the performance of an organization in project-based mission environments. In our model, organizations are constructed in terms of interacting components, namely, work and agents. The organizational structure depicts the grouping of agents, and the hierarchical arrangement of the groups. The congruence model of organizational behavior is based on the degree to which different components of the organization fit together. We use a discrete event simulator, specifically the Extend™ simulation package, to quantify the performance of an organization based on this model. We illustrate our approach using a symbolic example of an air operations center organization.

## **I. Introduction**

Organizational engineering is the process of configuring an organizational structure to accomplish a given high level task (termed a mission), while attempting to satisfy the stated performance objectives. An organization includes people supported by information-processing and communication tools.

Over the past forty years, simulation has become a primary tool for decision-making in engineering design (e.g., complex systems) and in discrete-event logistics systems (e.g., warehousing, manufacturing, and supply chains). Ostensibly, simulation models accomplish two valuable objectives: 1) they reveal, in a controlled way, the effects of interacting dynamics in complex systems, and 2) they create “synthetic histories”, which may reflect the impact of uncertainties in the occurrence of future events, for example, task demand. These synthetic histories can be studied to assess the impact of system design decisions, policies, decision algorithms, or ad hoc interventions. Because simulations are computational devices, many different synthetic histories, with different realizations of random processes, can be created, enabling quantitative risk assessment [1].

In many fields, including engineering, management and organizational science, simulations based on computational organization theory have been used to: (i) provide insight into the degree of match between the tasks and organizational structures, (ii) quantify how people, work processes and organizational structure influence the performance of tasks, (iii) identify bottlenecks, and (iv) improve the quality and efficiency of an organization [2]. Organizational simulation also provides an enabling toolkit for people to view, analyze, and to understand a current organization through interactive simulation, model the changes to an organization resulting from design and policy modifications and updates, and ascertain in a synthetic environment the intended and unintended effects of these changes.

Organization theory traditionally describes an organization only at an aggregate level, and provides general qualitative predictions about its overall behavior. Burton and Obel’s simple, but elegant, organization model [3] was more of a macro-contingency theory model. In this paper, we define and implement a “micro” model of the structure and behavior of the components of an organization, explicitly representing the work activities, groups of agents, and the organizational structure. This model is implemented using discrete event simulation to obtain quantitative estimates of the different metrics that reveal the efficiency of an organization.

Organizations rely on information for making decisions, controlling tasks and coordinating interrelated activities. To achieve these objectives, information must be transferred among the members of an organization via different means, for example, by face-to-face meetings, telephone conversations, memoranda and other novel methods enabled by modern technology (e.g., virtual collaborations). In this paper, we model information and knowledge as special types of resources, which can be acquired by people participating in certain tasks, e.g., information acquisition tasks. Knowledge and information can also be transferred among agents through communication. Structural contingency theory, and the literature on organizational design that has subsequently emerged based on it, is one of the most promising theoretical approaches to understanding organizational performance [3, 4, 8, 9]. Among the various derivatives of contingency theory, organization theorists have used the information-processing view of organizational behavior in a broad range of domains. Our model, following this view, depicts an organization as consisting of an information-processing and communication structure that is designed to achieve a specific set of goals, and comprised of individuals with limited capacity. It can be applied to complex, but relatively routine project-oriented design or planning tasks, found in many fields such as software development processes, military operations planning, and medical care procedures. To simulate the behavior of an organization working on a project with a concrete objective, we implement a symbolic work process and organization model using a discrete event simulator package, termed Extend™ [15]. We illustrate how the model implementation in Extend™ can be used to provide organization designers insights into the impact of the organizational structure on the activities, and ultimately its performance and effectiveness.

The rest of the paper is organized as follows: In Section II, our organizational model is presented, which includes the agents and their interrelationships, and the computational description of agent behavior. Section III provides an illustrative example to demonstrate the utility of the model in describing the structure of an organization and its underlying work environment, so as to monitor its activity, and to estimate its inherent performance. In Section IV, we conclude the paper and offer future research directions.

## II. Organizational model

A contemporary definition of an organization, given by Robbins [4], is as follows: *An organization is the planned coordination of the collective activities of two or more people who, functioning on a relatively continuous basis and through the division of labor and a hierarchy of authority, seek to achieve a common goal or a set of goals.* The key features of this definition, namely, “planned coordination, collective activity, division of labor, hierarchy, goals(s)” are all captured in the organizational model herein.

### 2.1 Organizational components

#### 2.1.1 Resource

We model the physical assets, knowledge, expertise, and information necessary for the processing of tasks as resources. The finite set of resources in the system is denoted by  $R = \{r_k = (r_{ck}, r_{qk}, r_{tk})\}_{k=1}^N$ , where  $r_{ck}$  is a unique identifier,  $r_{qk}$  is the quantity, and  $r_{tk}$  is the transferability indicator of resource  $r_k$ . If the transferability indicator is true, it implies that resource  $r_k$  possessed by a subset of the agents can be transferred (temporarily) to other agents through communication and coordination. We assume that the original agent will continue to own the resource after transfer. Some resources are dynamic, that is, an individual can learn and acquire these resources by working on some tasks, for example, an individual may acquire information, expertise or knowledge by working on a task.

#### 2.1.2 Work

Work is the basic activity in which an organization is engaged. The emphasis is on specific work activities that need to be performed and their inherent characteristics. Task analysis includes a description of the basic work activities and work flows. The work activities are denoted as tasks; each task may require a set of sub-tasks to be

completed to accomplish the task. The lowest level tasks require a set of resources to complete their execution accurately and with high quality.

Work flows, dependencies, and input-output relationships among tasks are encoded in a directed, acyclic graph, termed the task graph (see Fig. 1). The relationships among tasks considered in our model are ‘enable’, where a task cannot begin until all its enabling tasks are completed. The transitions among the task processes can be probabilistic. In the example task graph in Fig. 1(b),  $T_3$  (task 3 in Fig. 1(a)) is the enabling task of tasks  $T_4$  (task 1 as in Fig. 1(a)),  $T_5$  (task 3 as in Fig. 1(a)), and  $T_7$  (task 4 as in Fig. 1(a)). Depending on the outcome of the review process of task  $T_3$ , the task processes  $T_4$ ,  $T_5$ , or  $T_7$  may be executed.

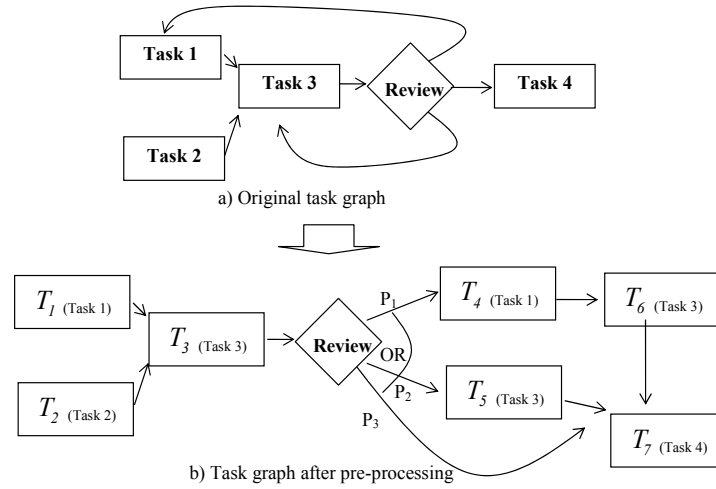


Fig. 1. Task graph illustration

A high level task may be decomposed into lower level subtasks, and this relationship is embodied in a subtask graph, as in Fig. 2.

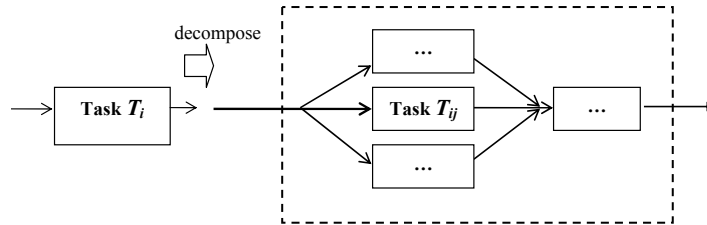


Fig. 2. Task to subtasks decomposition

Let  $T$  denote the set of all the lowest level tasks. Each lowest level task,  $T_i \in T$ , is characterized by the following attributes (see Fig. 3):

- a) required resources,  $Resources_R(T_i)$ : a vector of resources,  $Resources_R(T_i) \subseteq R$ ;
- b) baseline expected processing time:  $\overline{PT}(T_i)$ , which is the expected processing time of the task when all its resource requirements are satisfied;
- c) baseline workload per unit time:  $\overline{UWL}(T_i)$ , which is the expected workload of a task per unit time when all its resource requirements are satisfied;

- d) resources gained by an acting agent during task execution,  $Resource_G(T_i)$ ; these resources can be either transferable or non-transferable,  $Resource_G(T_i) \subseteq R$ ;
- e) synchronization task set,  $SYN(T_i)$ , the set of tasks that  $T_i$  needs to synchronize with during execution.

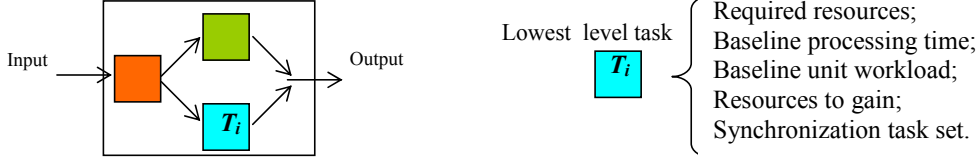


Fig. 3. Schematic depiction of requirements of a lowest level task

### 2.1.3 Agent

Agents, or decision makers (DM), either people or automatic programs, are the components of an organization, who complete the mission by performing the tasks. Agents engage in work by participating in different tasks, and play roles in the organization by virtue of being assigned to different organizational positions.

The most critical aspects of an agent that potentially influence its behavior include the knowledge, skills and years of experience, etc. Agents, as bounded rational actors [5], are cognitively limited, and therefore must join together to achieve a higher level of performance than what can be achieved by working on tasks without cooperation and coordination. Agents are also limited physically, both because of their physiology and the resources available to them, and therefore must coordinate their actions, e.g. to achieve higher-levels of productivity. The agent model takes these factors into account.

In our model, agents engage in the activities of an organization, by virtue of their task assignments. Each agent may be assigned to multiple tasks, while each lowest level task can be assigned to only one agent. The task processing capability of an individual is embodied into the individual's strategy, and cognitive constraints.

Let  $A$  denotes the set of agents. The characteristics of an agent,  $A_j \in A$ , include:

- a) resources: a set of resource items, i.e.,  $Resources(A_j) \subseteq R$ ; the agent can either own these resources a priori, can acquire them as a result of executing a task, or obtain them via resource transfers from other agents;
- b) maximum unit workload,  $Max\_UWL(A_j)$ . We assume that, at any time, the workload of an agent cannot exceed its maximum unit workload;
- c) task assignments,  $Tasks(A_j) \subset T$ , i.e., the set of lowest level tasks assigned to agent  $A_j$ . Each agent is fully aware of the status of its tasks. In our model, each lowest level task is executed by one and only one agent, i.e.,  $Tasks(A_{j_1}) \cap Tasks(A_{j_2}) = \emptyset$ , when  $A_{j_1} \neq A_{j_2}$ .

### 2.1.4 Organization

Organizational structure includes the partitioning of people into groups and the hierarchical arrangement of the groups establishing clear lines of responsibility. Due to differences among resources possessed by the agents and those required by the tasks, the structure will determine the organizational activity, and directly impact the organizational performance, for example, mission completion time and quality, and workload distribution of agents. The influence of a particular organizational structure on mission completion can be captured in terms of different parameters, including the communication pattern, load and latency. Different organizational structures, which constitute different hierarchical arrangements and divisions, would exhibit varying elements of these factors.

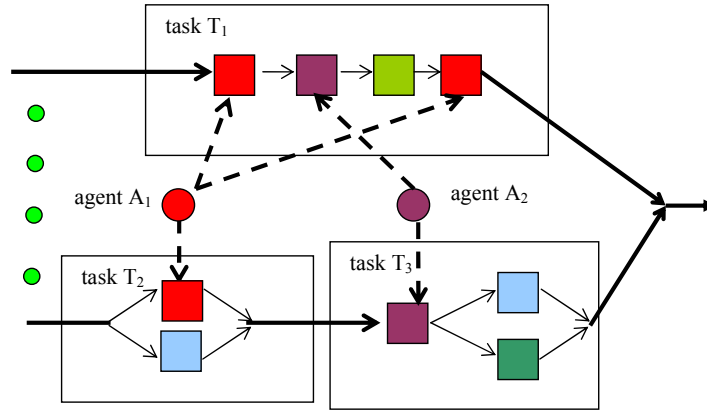


Fig 4. Assignment of agents to tasks

If an agent is working on a task, which requires transferable resources, the agent may request these resources from other agents who own them. For better situational awareness, e.g., when an agent needs to know the status of the tasks of other agents or the workloads of some agents, it may communicate with other agents to retrieve this information. For processing synchronized tasks, agents will coordinate with other agents to schedule the tasks that need to be processed simultaneously. Therefore, the organizational structure, which determines the communication channels among agents, as in Fig.5, directly influences the communication effectiveness of an organization.

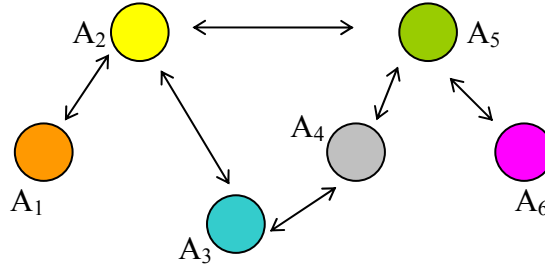


Fig.5. Communication channels among agents

There are two different types of communication considered in the model:

- Coordination, which occurs when agents engage in synchronizing tasks;
- Communication, which occurs when a transfer of resources is necessary, and when agents inform other agents of the status of some tasks.

We assume that two agents, who share a direct communication channel, will communicate or coordinate without any intermediate agents, for example, agents  $A_1$  and  $A_2$ , or agents  $A_5$  and  $A_6$  in Fig. 5. When no direct channel exists among two agents, they will choose a path with a fewest number of intermediate agents, e.g., agents  $A_1$  and  $A_3$ , will communicate and coordinate via agent  $A_2$ .

## 2.2 Interrelationships among organizational components

### 2.2.1 Performance measures

The interdependencies among the components of an organization determine its performance. We consider the following metrics to evaluate organizational performance:

- Mission completion time;
- Mission quality, and qualities (accuracies) of individual tasks;
- Workload distribution among the agents; and
- Communication and coordination load of individual agents and the organization.

A relative degree of congruence exists between each pair of organizational components, which is defined as the degree to which the needs, demands, goals, and structures of one component are consistent with the needs, demands, goals, and structures of another component [6,7]. Congruence is therefore a measure of how well pairs of components fit together.

The aggregate model, or the organization as a whole, displays a relatively high (low) degree of congruence in that pairs of components have high (low) congruence. The basic hypothesis of the model is as follows: other things being equal, the greater the total degree of congruence among the various components, the more effective will be the organization, where effectiveness is defined as the degree of closeness between the actual performance of the organization to the expected or planned performance as specified by the organizational strategy.

### 2.2.2 Agents to tasks' assignment

When an agent is assigned to a lowest level task, the following performance measures are considered:

- Quality of the task performed by the agent;
- Actual processing time of the task;
- Actual workload imposed on the agent.

For an assignment of agent  $A_j$  to task  $T_i$ , the processing time, quality and workload of the task will be determined as follows:

$$t\_process\_time(T_i, A_j) = PT(\overline{PT}(T_i), Resources_R(T_i), Resources(A_j));$$

$$t\_quality(T_i, A_j) = Q(Resources_R(T_i), Resources(A_j)); \text{ and}$$

$t\_UWL(T_i, A_j) = UWL(\overline{UWL}(T_i), Resources_R(T_i), Resources(A_j))$ , here  $PT$ ,  $Q$  and  $UWL$  are monotonic functions to embody the congruence theory. Possible examples of these functions are:

$$PT(\overline{PT}(T_i), Resources_R(T_i), Resources(A_j)) = \overline{PT}(T_i) \prod_{r_k(T_i) \in R_R(T_i)} \left[ U_{PT} \left( \frac{r_{qk}(A_j)}{r_{qk}(T_i)} \right) + N(0, s_{PT}) \right];$$

$$Q(Resources_R(T_i), Resources(A_j)) = \prod_{r_k(T_i) \in R_R(T_i)} \left[ U_Q \left( \frac{r_{qk}(A_j)}{r_{qk}(T_i)} \right) + N(0, s_Q) \right];$$

$$UWL(\overline{UWL}(T_i), Resources_R(T_i), Resources(A_j)) = \overline{UWL}(T_i) \prod_{r_k(T_i) \in R_R(T_i)} \left[ U_{WL} \left( \frac{r_{qk}(A_j)}{r_{qk}(T_i)} \right) + N(0, s_{WL}) \right].$$

Here  $U_{PT}$ ,  $U_Q$  and  $U_{WL}$ , as in Fig. 6., are monotonic functions of the ratio of the quantity of a resource owned by an agent to the quantity of a resource required by a task.  $N(0, \sigma)$  denotes a normal random variable, with mean 0 and standard deviation  $\sigma$ . Fig. 6 shows that when the quantity of a resource possessed by an agent is sufficient, it incurs less processing time, completes the task with higher quality, and at a lower workload. As the resource



quantity of the agent compared to what is required by the task decreases, the processing time, quality and workload deteriorate. For real world cases, functions  $PT$ ,  $Q$  and  $UWL$  are gleaned from subject matter experts.

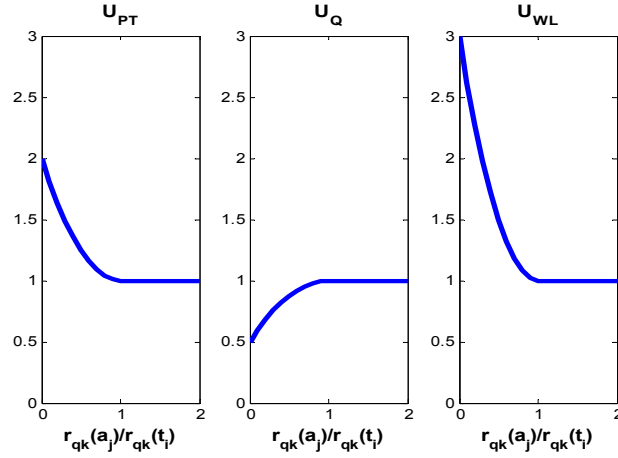


Fig. 6. Monotonic function examples for  $PT$ ,  $Q$  and  $UWL$

### 2.2.3 Dependence among tasks

#### a) Tasks at the same level

- Timing interdependence, i.e., an enabled task cannot begin until all its enabling tasks are completed.
- Quality interdependence, i.e., the quality of the output of enabling tasks will influence the quality of an enabled task.

#### b) Subtasks and high level tasks

The performance of a high-level task depends on the performance of its composite subtasks:

- Quality of the high-level task versus the quality of the composite subtasks;
- Processing time of the high-level task versus the processing time of the composite subtasks.

The quality of a high-level task depends on the quality of its composite subtasks. If the performance of the subtasks is poor, it will lead to a poor performance of the high-level task. On the other hand, if the performance of subtasks is superior, it will facilitate better performance of the high-level task. Such relations could be formalized as follows:  $Q(T_i) = \min_{T_{ij} \in T_i} (Q(T_{ij}))$  or  $Q(T_i) = \text{avg}_{T_{ij} \in T_i} (Q(T_{ij}))$ , where  $T_{ij}$  is a subtask of parent task  $T_i$ . Many other

functional variants to compose the quality of a high-level task from the quality of its subtasks are possible. These functions also need to be elicited from subject matter experts.

### 2.3 Agent behavior

The ability of an organization to fulfill a mission is dependent on the intelligence of the agents comprising the organization. Agent based model (ABM) of human behavior is a useful tool because modeling human behavior potentially involves taking into account many factors and partial theories, which can be integrated within an ABM to see how well they hold together [10, 11]. Additionally, it is easier to explore the dynamics of a phenomenon with an ABM.

At any time, we assume that an agent knows the status of all its own tasks. The status of a lowest level task can be any one of the following:

- Not ready, when at least one of its enabling tasks is not completed;
- Ready, when all its enabling tasks are completed, but it has not yet been scheduled;
- Scheduled, when it has been scheduled by its owner agent for a specific starting time;
- Working;

- Completed.

Three types of communication messages can be exchanged among the agents:

- $COMM1(A_j, A_k, T_i)$ : Agent  $A_j$  sends a message to agent  $A_k$  to notify that a task  $T_i$  is completed;
- $COMM2(A_j, A_k, R_T)$ : Agent  $A_j$  requests resource(s)  $R_T \subseteq R$  from agent  $A_k$  ;
- $COMM3(A_j, A_k, T_h, ST)$ : Agent  $A_j$  sends a synchronization request to agent  $A_k$  to start the task  $T_h$  at time  $ST$  .

When a message is exchanged among two agents, each party incurs a communication load.

We assume that an agent is proactive and responsible in scheduling the tasks, as long as its workload constraint is satisfied. When an agent finishes a task, it will collect the information necessary to schedule the next task, such that the projected mission completion time is the shortest. It will also coordinate with other agents to synchronize tasks cooperatively, as long as the projected workload is within its workload constraints.

Prior to working on a task, if the task needs a transferable resource that is not owned by the agent, it selects another agent who owns the resource and requests a resource transfer. The selection is based on the shortest communication path rule described in Section 2.1.4.

An agent  $A_j$ , at any time, maintains the following information:

- $TODO(A_j)$ : A list of tasks the agent needs to perform, which the agent hasn't begun to do yet;
- $READY(A_j)$ : A list of tasks, such that for each task all the enabling tasks are completed, while they haven't begun yet;
- $SCHEDULED(A_j)$ : A list of tasks which have not yet begun, but their start times are scheduled;
- $DOING(A_j)$ : A list of tasks the agent is working on;
- $IWL(A_j, t)$ : Current instant workload of agent  $A_j$  at time  $t$  .

#### **Scheduling procedure for Agent $A_j$ :**

**Repeat** the following two steps until  $READY(A_j) = \emptyset$  or any new task to be scheduled at the current time would make  $IWL(A_j, t)$  exceed  $Max\_UWL(A_j)$ , for some  $t \geq t'$ , where  $t'$  is the current time.

**Step 1:** For  $\forall T_k \in READY(A_j)$ , calculate

$$t\_process\_time(T_k, A_j) = PT(\overline{PT}(T_k), Resources_R(T_k), Resources(A_j));$$

$$t\_quality(T_k, A_j) = Q(Resources_R(T_k), Resources(A_j)); \text{ and}$$

$$t\_UWL(T_k, A_j) = UWL(Resources_R(T_k), Resources(A_j)); \text{ and estimate the mission completion time (MCT),}$$

i.e.,  $\widehat{MCT}_k$ , by computing the expected mission completion time of each possible mission path, weighted by their probabilities.

Select a task  $T_i \in READY(A_j)$ , such that  $\widehat{MCT}_i$  is as low as possible, and

$IWL(A_j, t) + t\_UWL(T_i, A_j) < Max\_UWL(A_j)$ , for  $t' < t < t' + t\_process\_time(T_i, A_j)$ , where  $t'$  is the current time.

**Step 2:**

*Step 2.1:* Agent  $A_j$  checks the resource requirements of task  $T_i$ . For every transferable resource  $R_T (\subset R)$  needed by task  $T_i$ ,  $A_j$  identifies the agent, say,  $A_k$  which has the resource, and the communication path to reach  $A_k$ . It then sends along the path a request  $COMM2(A_j, A_k, R_T)$  to agent  $A_k$ .

When all the transferable resources needed by task  $T_i$  have been acquired by agent  $A_j$ , remove  $T_i$  from  $READY(A_j)$ ; add  $T_i$  to  $DOING(A_j)$ , and set its processing time to  $t\_process\_time(T_i, A_j)$ ;

*Step 2.2:* Start the task execution, update the instant workload according to:

$IWL(A_j, t) = IWL(A_j, t) + t\_UWL(T_i, A_j)$ ,  $t' \leq t < t' + t\_process\_time(T_i, A_j)$ ; and save the record  $\{A_j, T_i, Start\_Time, t\_process\_time, t\_quality, t\_UWL\}$ .

*Step 2.3:* If the task  $T_i$  is a synchronized task, agent  $A_j$  identifies all the tasks, for example,  $T_h$ , that belong to  $SYN(T_i)$ , and their owner agents, for example  $A_k$ . Agent  $A_j$  sends the owner agents a message  $COMM3(A_j, A_k, T_h, t' + \Delta)$ , where  $\Delta$  is the total time incurred in communication and while waiting to receive acknowledgements from the corresponding agents. When the acknowledgements of  $COMM3(A_j, A_k, T_h, t' + \Delta)$  are received, agent  $A_j$  schedules the begin time of  $T_i$  as  $t' + \Delta$ , and moves  $T_i$  from  $READY(A_j)$  to  $SCHEDULED(A_j)$ .

**Agent  $A_j$  event handling procedure (at time  $t'$ ):**

**Case 1:** When task  $T_i$  is completed, agent  $A_j$  deletes task  $T_i$  from  $DOING(A_j)$ ; updates  $Resources(A_j)$  by adding  $Resource_G(T_i)$ ; transfers any  $T_i$ 's enabling tasks owned by agent  $A_j$  from  $TODO(A_j)$  to  $READY(A_j)$ ; updates  $IWL(A_j, t) = IWL(A_j, t) - t\_UWL(T_i, A_j)$ ,  $t \geq t'$ ; sends the owners, e.g.,  $A_k$ , of all  $T_i$ 's enabling tasks a message  $COMM1(A_j, A_k, T_i)$ , and applies the scheduling procedure to schedule the next task that is ready;

**Case 2:** Agent  $A_j$  starts tasks in  $SCHEDULED(A_j)$  list at the times scheduled; updates its instantaneous workload and saves the record correspondingly;

**Case 3:** When an agent  $A_j$  receives a message  $COMM2(A_k, A_j, R_T)$ , it transfers the requested resource  $R_T$  to  $A_k$  by sending a message  $COMM2\_ACK(A_j, A_k, R_T)$ ;

**Case 4:** When an agent  $A_j$  receives a synchronization request  $COMM3(A_k, A_j, T_h, ST)$  from agent  $A_k$ , it checks its availability. If it is available, it sends agent  $A_k$  a message  $COMM3\_ACK(A_k, A_j, T_h, ST)$ . If it is not available, it sends agent  $A_k$  a message  $COMM3\_NAK(A_k, T_h, new\_ST)$ , where  $new\_ST$  is the new start time proposed by agent  $A_k$ .

## 2.4 Implementation

We use the discrete event simulator encapsulated in the Extend<sup>TM</sup> software [15] package to implement the organizational model described in Sections 2.1 through 2.3. Extend<sup>TM</sup> incorporates a full array of building blocks from continuous, stochastic and discrete event system domains, and allows unlimited hierarchical decomposition to enable building and understanding of complex systems. Furthermore, Extend<sup>TM</sup> provides a built-in, compiled C-like programming language, to enable users to build their own libraries and customize blocks to their needs.

The input and output products (e.g., documents) of each task in our model are simulated as discrete events. Tasks at various levels of abstraction are encoded in the hierarchies of task blocks. The logical dependencies among tasks, i.e., sequential, conditional, optional, or concurrent, are implemented via the existing blocks in Extend<sup>TM</sup>, namely, select, decision, activity, multiplex, and batch blocks of the discrete event library. The scheduling strategies of agents, the controlling factors in the task environment, are implemented via self-designed blocks. Uncertainties in the task environment and organizational behavior are modeled by random and statistical blocks of Extend<sup>TM</sup> generic library.

## III. Illustrative example

We use a symbolic example of an Air Operations Center (AOC) organization to illustrate our modeling and simulation approach. The AOC [16] is composed of two divisions, namely, information operations division and kinetic operations division. Information operations division is composed of information operations officers 1-4, (IOO1-IOO4); and kinetic operations division is composed of kinetic operations officers 1-4, (KOO1-KOO4), as in Fig. 7 (a). The task graph, as depicted in Fig. 8(a), has three task chains, each with a different start time. Each chain represents a production process of an Air Task Order (ATO). Each task chain has three major tasks in a sequence, i.e., JIPTL (Joint Integrated Prioritized Target List Production), MAAP (Master Air Attack Plan Production), and ATO. The detailed task graph for each task chain is depicted in Fig. 8(b). The model parameters, namely, the configuration of resources, tasks, and agents are listed in Tables 1, 2 and 3 respectively.

In this simulation example, we begin with an organization structure, namely, ORG1. Its communication structures and assignment of agents to tasks are reported in Fig. 7(b) and Table 4, respectively.

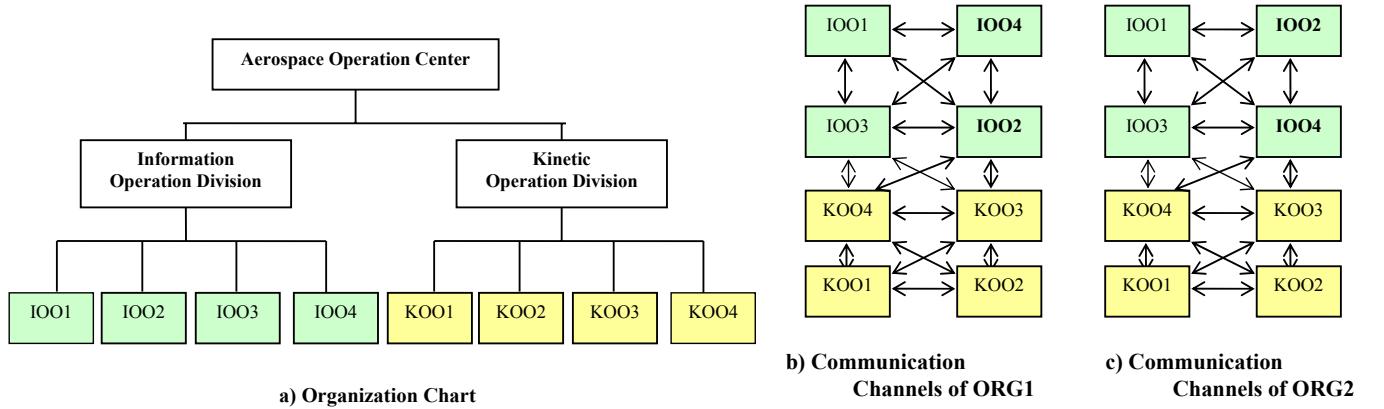


Fig. 7. Organization structure of CAOC

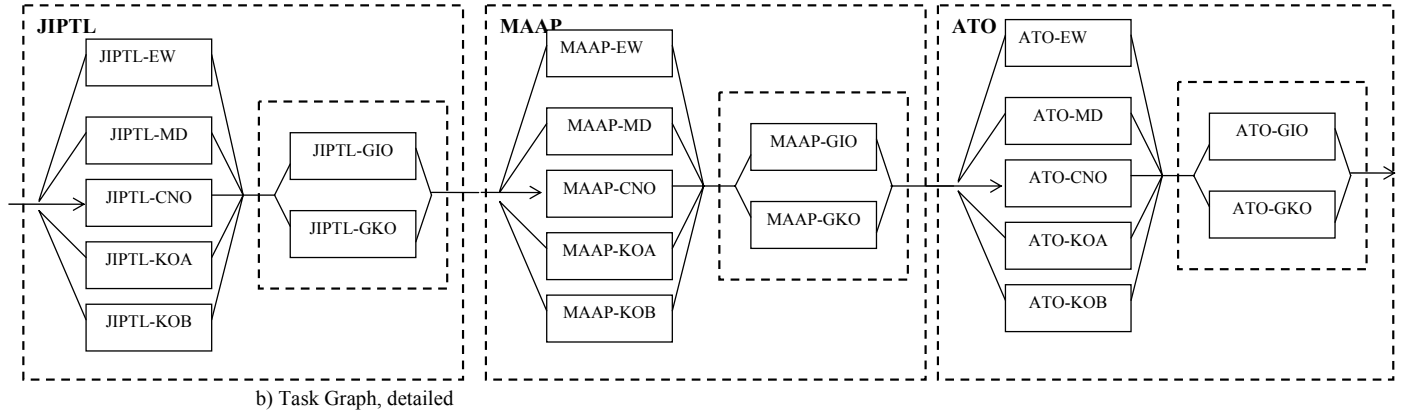
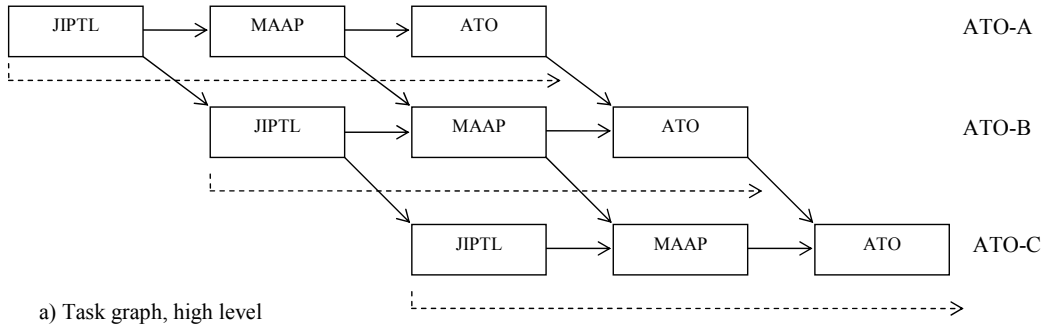


Fig. 8. AOC task graph

Resource Name		T*	Resource Name		T*
Electronic warfare expertise	EW	NO	EW-JIPTL information	EW-JIPTL	YES
Military deception expertise	MD	NO	MD-JIPTL information	MD-JIPTL	YES
Computer network operation expertise	CNO	NO	CNO-JIPTL information	CNO-JIPTL	YES
General information operation expertise	GIO	NO	KOA-JIPTL information	KOA-JIPTL	YES
Kinetic operation A expertise	KOA	NO	KOB-JIPTL information	KOB-JIPTL	YES
Kinetic operation B expertise	KOB	NO	EW-MAAP information	EW-MAAP	YES
General kinetic operation expertise	GKO	NO	MD-MAAP information	MD-MAAP	YES
Communication expertise	COMM	NO	CNO-MAAP information	CNO-MAAP	YES
JIPTL expertise	JIPTL	NO	KOA-MAAP information	KOA-MAAP	YES
MAAP expertise	MAAP	NO	KOB-MAAP information	KOB-MAAP	YES
ATO expertise	ATO	NO	EW-ATO information	EW-ATO	YES
			MD-ATO information	MD-ATO	YES
JIPTL information	JIPTL-I	YES	CNO-ATO information	CNO-ATO	YES
MAAP information	MAAP-I	YES	KOA-ATO information	KOA-ATO	YES
ATO information	ATO-I	YES	KOB-ATO information	KOB-ATO	YES

Table 1. Resources in AOC example (T\* is for transferability)

Task	Lowest Level Task	UWL (SYN)	PT	Required Resource Name (quantity)	Gained Resource
Joint Integrated Prioritized Target List (JIPTL) production	JIPTL-EW	10	10	JIPTL(2) EW(2) GIO(1) JIPTL-I*(1)	EW-JIPTL (2)
	JIPTL-MD	10	10	JIPTL(2) MD(2) GIO(1) JIPTL-I*(1)	MD-JIPTL (2)
	JIPTL-CNO	10	10	JIPTL(2) CNO(2) GIO(1) JIPTL-I*(1)	CNO-JIPTL (2)
	JIPTL-KOA	10	10	JIPTL(2) KOA(2) GKO(1) JIPTL-I*(1)	KOA-JIPTL (2)
	JIPTL-KOB	10	10	JIPTL(2) KOB(2) GKO(1) JIPTL-I*(1)	KOB-JIPTL (2)
	JIPTL-GIO	5 JIPTL-GKO	5	JIPTL(1) EW-JIPTL(1) MD-JIPTL(1) CNO-JIPTL(1) COMM(2)	JIPTL-I (1)
	JIPTL-GKO	5 JIPTL-GIO	5	JIPTL(1) KOA-JIPTL(1) KOB-JIPTL(1) COMM(2)	JIPTL-I (1)
Master Air Attack Plan (MAAP) production	MAAP-EW	10	10	MAAP(2) EW(2) GIO(1) JIPT-I(1) MAAP-I*(1)	EW-MAAP (2)
	MAAP-MD	10	10	MAAP(2) MD(2) GIO(1) JIPT-I(1) MAAP-I*(1)	MD-MAAP (2)
	MAAP-CNO	10	10	MAAP(2) CNO(2) GIO(1) JIPT-I(1) MAAP-I*(1)	CNO-MAAP (2)
	MAAP-KOA	10	10	MAAP(2) KOA(2) GKO(1) JIPT-I(1) MAAP-I*(1)	KOA-MAAP (2)
	MAAP-KOB	10	10	MAAP(2) KOB(2) GKO(1) JIPT-I(1) MAAP-I*(1)	KOB-MAAP (2)
	MAAP-GIO	5 MAAP-GKO	5	MAAP(1) EW-MAAP(1) MD-MAAP(1) CNO-MAAP(1) COMM(2)	MAAP-I (1)
	MAAP-GKO	5 MAAP-GIO	5	MAAP(1) KOA-MAAP(1) KOB-MAAP(1) COMM(2)	MAAP-I (1)
Air Task Order (ATO) production	ATO-EW	10	10	ATO(2) EW(2) GIO(1) MAAP-I(2) ATO-I*(1)	EW-ATO (2)
	ATO-MD	10	10	ATO(2) MD(2) GIO(1) MAAP-I(2) ATO-I*(1)	MD-ATO (2)
	ATO-CNO	10	10	ATO(2) CNO(2) GIO(1) MAAP-I(2) ATO-I*(1)	CNO-ATO (2)
	ATO-KOA	10	10	ATO(2) KOA(2) GKO(1) MAAP-I(2) ATO-I*(1)	KOA-ATO (2)
	ATO-KOB	10	10	ATO(2) KOB(2) GKO(1) MAAP-I(2) ATO-I*(1)	KOB-ATO (2)
	ATO-GIO	5 ATO-GKO	5	ATO(1) EW-ATO(1) MD-ATO(1) CNO-ATO(1) COMM(2)	ATO-I (1)
	ATO-GKO	5 ATO-GIO	5	ATO(1) KOA-ATO(1) KOB-ATO(1) COMM(2)	ATO-I (1)

Table 2. Tasks in AOC example (\* means resources from previous task chain)

Division	Agent	Max_UWL	Qualification Resources
Information Operation Division	IOO1	30	EW(2) MD(1) CNO(1) GIO(1) JIPTL(2) MAAP(2) ATO(1)
	IOO2	30	EW(1) MD(2) CNO(1) GIO(1) JIPTL(1) MAAP(2) ATO(2)
	IOO3	30	EW(1) MD(1) CNO(2) GIO(1) COMM(2) JIPTL(1) MAAP(1) ATO(2)
	IOO4	30	EW(1) MD(1) CNO(1) GIO(2) COMM(2) JIPTL(1) MAAP(1) ATO(1)
Kinetic Operation	KOO1	30	KOA(2) KOB(1) GKO(1) COMM(1) JIPTL(2) MAAP(1) ATO(1)
	KOO2	30	KOA(1) KOB(2) GKO(1) COMM(1) JIPTL(1) MAAP(2) ATO(2)

Division	KOO3	30	KOA(1) KOB(1) GKO(2) COMM(2) JIPTL(1) MAAP(1) ATO(1)
	KOO4	30	KOA(1) KOB(1) GKO(1) COMM(2) JIPTL(1) MAAP(2) ATO(2)

Table 3. Agents in AOC example

Task Chain	Task	ORG1	ORG2	Task	ORG1	ORG2
ATO-A	JIPTL-EW	IOO1	IOO1	MAAP-KO2	KOO2	KOO2
	JIPTL-MD	IOO2	IOO2	MAAP-GIO	IOO4	IOO4
	JIPTL-CNO	IOO3	IOO3	MAAP-GKO	KOO3	KOO4
	JIPTL-KOA	KOO1	KOO1	ATO-EW	IOO4	IOO1
	JIPTL-KOB	KOO3	KOO2	ATO-MD	IOO1	IOO2
	JIPTL-GIO	IOO4	IOO4	ATO-CNO	IOO3	IOO3
	JIPTL-GKO	KOO4	KOO3	ATO-KOA	KOO4	KOO1
	MAAP-EW	IOO1	IOO1	ATO-KOB	KOO2	KOO2
	MAAP-MD	IOO2	IOO2	ATO-GIO	IOO4	IOO4
	MAAP-CNO	IOO4	IOO3	ATO-GKO	KOO3	KOO3
	MAAP-KOA	KOO1	KOO1			
ATO-B	JIPTL-EW	IOO1	IOO1	MAAP-KO2	KOO3	KOO2
	JIPTL-MD	IOO2	IOO2	MAAP-GIO	IOO4	IOO4
	JIPTL-CNO	IOO3	IOO3	MAAP-GKO	KOO3	KOO3
	JIPTL-KO1	KOO1	KOO1	ATO-EW	IOO1	IOO1
	JIPTL-KO2	KOO2	KOO2	ATO-MD	IOO3	IOO2
	JIPTL-GIO	IOO4	IOO4	ATO-CNO	IOO4	IOO3
	JIPTL-GKO	KOO4	KOO4	ATO-KOA	KOO4	KOO1
	MAAP-EW	IOO1	IOO1	ATO-KOB	KOO2	KOO2
	MAAP-MD	IOO3	IOO2	ATO-GIO	IOO4	IOO4
	MAAP-CNO	IOO4	IOO3	ATO-GKO	KOO3	KOO4
	MAAP-KOA	KOO1	KOO1			
ATO-C	JIPTL-EW	IOO1	IOO1	MAAP-KOB	KOO2	KOO2
	JIPTL-MD	IOO2	IOO2	MAAP-GIO	IOO4	IOO4
	JIPTL-CNO	IOO4	IOO3	MAAP-GKO	KOO3	KOO4
	JIPTL-KOA	KOO1	KOO1	ATO-EW	IOO4	IOO1
	JIPTL-KOB	KOO3	KOO2	ATO-MD	IOO3	IOO2
	JIPTL-GIO	IOO3	IOO3	ATO-CNO	IOO2	IOO3
	JIPTL-GKO	KOO4	KOO3	ATO-KOA	KOO1	KOO1
	MAAP-EW	IOO4	IOO1	ATO-KOB	KOO4	KOO2
	MAAP-MD	IOO2	IOO2	ATO-GIO	IOO4	IOO4
	MAAP-CNO	IOO3	IOO3	ATO-GKO	KOO3	KOO3
	MAAP-KOA	KOO1	KOO1			

Table 4. Organizational assignment in AOC example

The results of the ORG1 structure obtained by simulating the AOC organization using the model implemented in Extend™ are shown in Figs. 9, 10, 11 and 12. The performance measures, namely, qualities of tasks, workload and communication load of agents, and mission completion time of each task chain are obtained from 100 Monte-Carlo runs.

From the results of ORG1's simulation, organization designers can gain the following insights: (i) the designated organizational structure and assignments lead to comparatively low qualities for some tasks, e.g., MAAP-CNO(ID=10) of task chain ATO-A, ATO-CNO (ID=17) of task chain ATO-B, MAAP-KOA (ID=8) of task chain ATO-C; (ii) agent IOO4 is under considerably high workload, i.e., 2.4 times of the average of workload of the other agents; (iii) agents IOO2, IOO4, KOO3, KOO4 incur substantial communication and coordination load, and (iv) task chain ATO-A would be completed in lesser time than ATO-B and ATO-C on an average. Accordingly, the designer could adjust the organizational structure or the assignment, so as to meet the organization objectives.

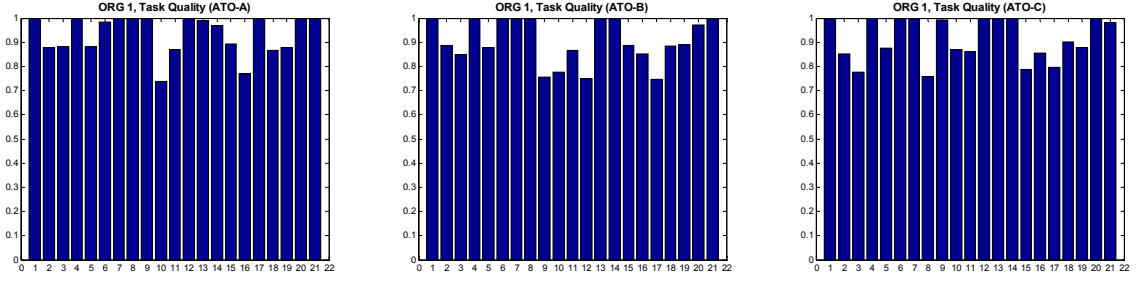


Fig. 9. Quality of Tasks Completion of ORG1 in AOC example

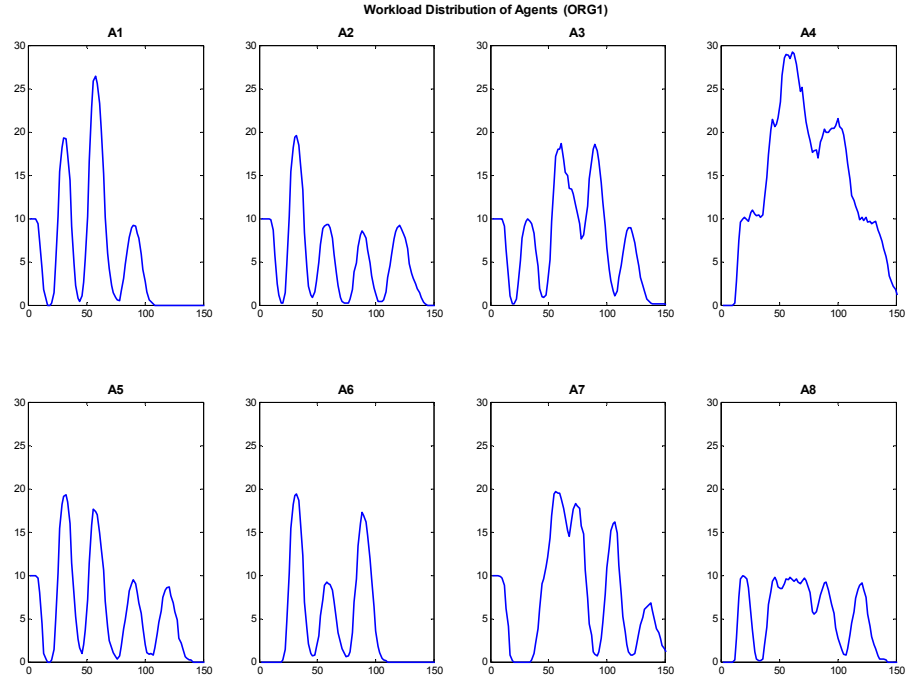


Fig. 10. Workload distribution of ORG1 in AOC Example



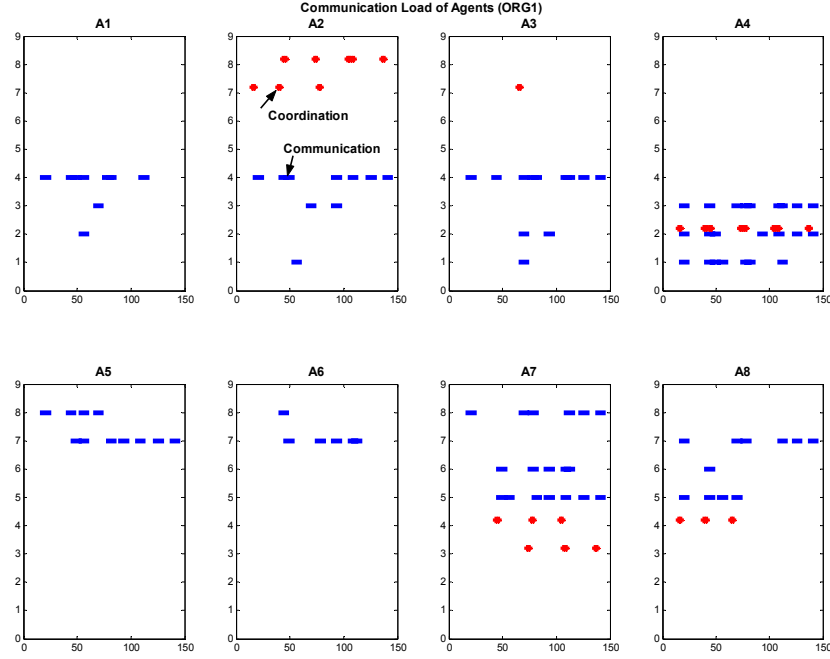


Fig. 11. Communication load distribution of ORG1 in AOC example

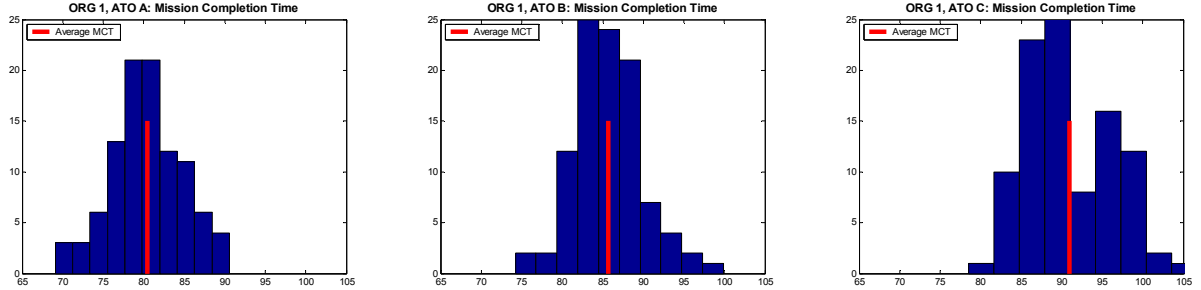


Fig. 12. Task chain completion times comparison of ORG1 in AOC example

Altering the organization structure from ORG1 to ORG2, as defined in Fig. 7(c) and Table 4, the designer can obtain the organizational performance based on 100 Monte-Carlo runs as shown in Fig. 13 to Fig. 16. The following performance improvements are observed: (i) higher qualities of task completion on an average, and with less variance in the quality among tasks; (ii) better balance in the workload among agents; (iii) agent IOO2 has less coordination load, and IOO4, who is engaged in coordinated task assignments, is more efficient at coordinating in ORG2 than in ORG1; (iv) the missions of ATO-A, ATO-B and ATO-C can all be completed in significantly less time.

To corroborate that ORG2's performance improvement is statistically significant, the organization designer can suggest and test the hypotheses of the following type:

$H_{10}$ : ORG2 and ORG1 have similar average task qualities;

$H_{1a}$ : ORG2 has higher average task quality than ORG1;

$H_{20}$ : ORG2 and ORG1 have similar mission completion times;

$H_{2a}$ : ORG2 has less average mission completion time than ORG1;

$H_{30}$ : ORG2 and ORG1 have similar average agent workload;

$H_{3a}$ : ORG2 has less average agent workload than ORG1;

$H_{40}$ : ORG2 and ORG1 have similar variance in the agent workload;

$H_{4a}$ : ORG2 has less agent workload variance than ORG1;

We ran 1000 Monte-Carlo runs of ORG1 and ORG2 to test these hypotheses. A  $t$ -test statistic [17] defined as

$\frac{\bar{Y}_2 - \bar{Y}_1}{\sqrt{S_1^2 / N_1 + S_2^2 / N_2}}$  is applied, here  $\bar{Y}_1$  and  $\bar{Y}_2$  are the sample means of some performance measure from ORG1

and ORG2 respectively,  $S_1^2$  and  $S_2^2$  are the sample variances from ORG1 and ORG2 respectively, and  $N_1$  and  $N_2$  are the sample size for ORG1 and ORG2 respectively. For hypothesis  $H_{10}/H_{1a}$ , from 1000 samples of average task quality, we found that ORG1 has a mean of 0.917 and a standard deviation of 0.022, while ORG2 has a mean of 0.946 and a standard deviation of 0.023. The  $t$ -test statistic of  $H_{10}$  is 20.3 ( $P(t \leq -3.29) = P(t \geq 3.29) = 0.0005$ , for  $t$  distribution with 999 degrees of freedom). Therefore this result is significant at 0.1% level and beyond, indicating that hypothesis  $H_{10}$  can be rejected with confidence and  $H_{1a}$  is strongly preferred.

From 1000 samples of the average mission completion time of task chains, ORG1 has a mean of 86.3 units and a standard deviation of 4.2 units, while ORG2 has a mean of 79.18 and a standard deviation of 3.4 units; therefore  $t$  test statistic (with 999 degrees of freedom) of  $H_{20}$  is - 29.6, implying that  $H_{20}$  can be rejected with confidence and  $H_{2a}$  is strongly preferred.

From 1000 samples of average workload among agents, ORG1 has a mean of 930.4 and a standard deviation of 22.3 units, while ORG2 has a mean of 870.7 units and a standard deviation of 21.3 units; therefore the  $t$ -test statistic of  $H_{30}$  is - 43.3 (with 999 degrees of freedom), indicating that  $H_{30}$  can be rejected with confidence and  $H_{3a}$  is strongly preferred.

From 1000 samples of the variance of workload among agents, ORG1 has a mean of 393.8 units and a standard deviation of 31.6 units, while ORG2 has a mean of 315.2 units and a standard deviation of 21.1 units; therefore the  $t$ -test statistics (with 999 degrees of freedom) of  $H_{40}$  is - 47.1, indicating  $H_{40}$  can be rejected with confidence and  $H_{4a}$  is strongly preferred.

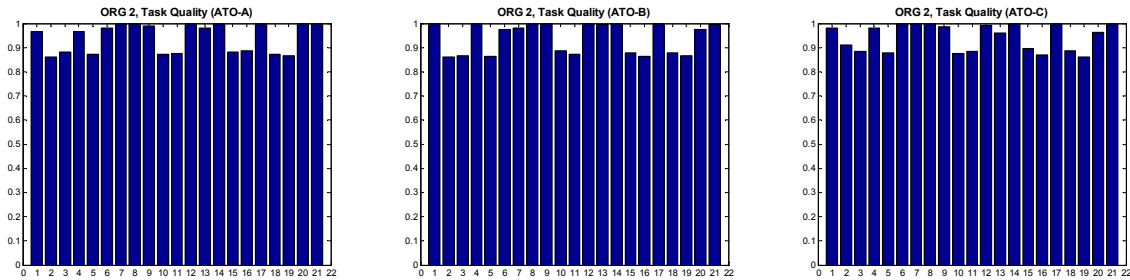


Fig. 13. Quality of Tasks Completion of ORG2 in AOC example

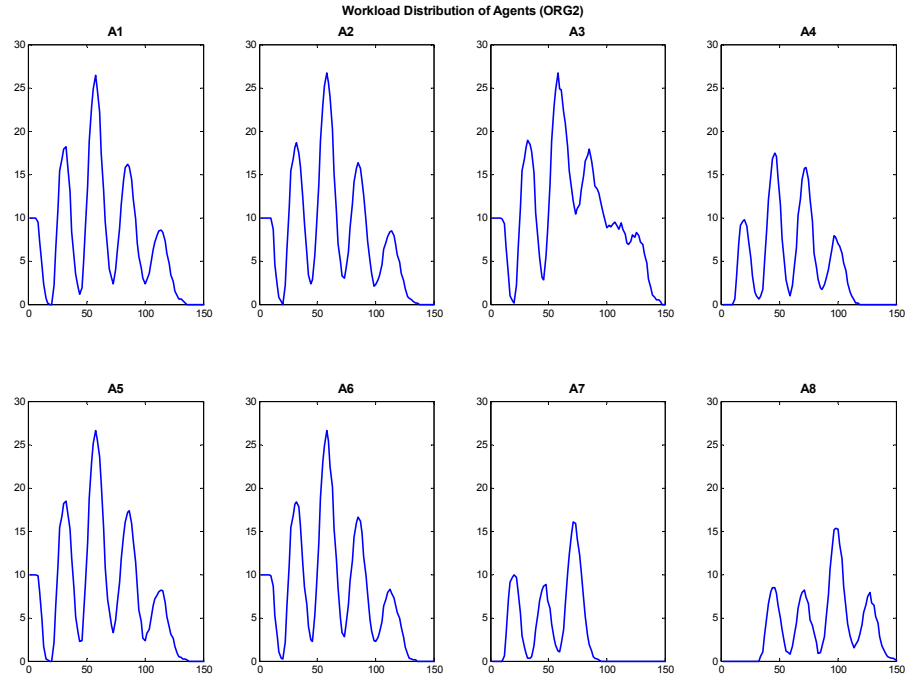


Fig. 14. Workload distribution of ORG2 in AOC Example

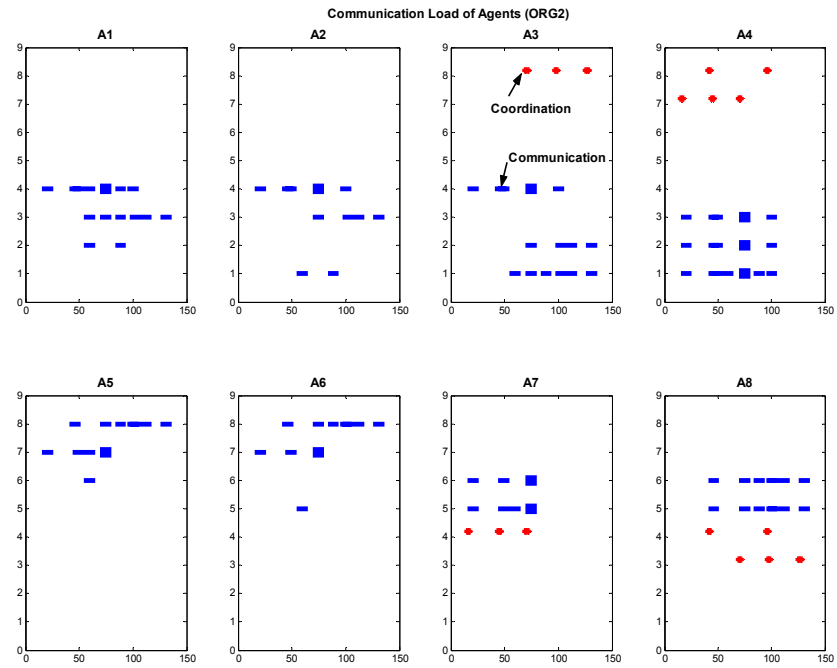


Fig. 15. Communication load distribution of ORG2 in AOC example

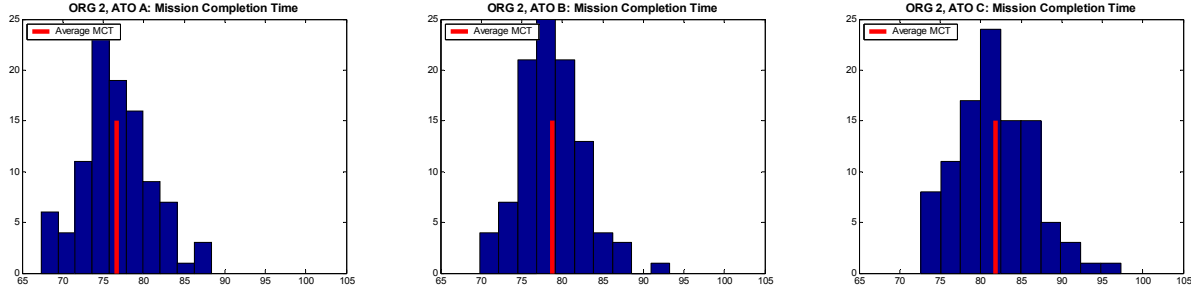


Fig. 16. Task chain completion times comparison of ORG2 in AOC example

#### IV. Summary

In this paper, we proposed a modeling and simulation methodology for organizations involved in project-based missions. In our model, organizations are constructed in terms of interacting components, namely, work, agents and the organizational structure, which depicts the assignment of work to agents, and the hierarchical arrangement of agents. The effectiveness of an organization reflects the congruence of the above three components. The information, expertise and knowledge are modeled as dynamic resources in the system, where agents can acquire these resources by working on tasks, and also by communicating with other agents. The organizational model is implemented using the Extend™ simulation package, which embodies a discrete event simulator. We also presented a symbolic example of an air operations center organization to illustrate the potential of our modeling and simulation approach.

In our future work, we propose to consider more realistic and full-rang of task interrelationships, sophisticated agent behavior model, and the impact of agent behavior on the task completion model, modeling of information integration and dissemination along organizational hierarchies, and errors in information propagation.

#### References

- [1] D. Nadler (1992), *Organizational Architecture: Designs for changing organizations*, Jossey-Bass Publishers, San Francisco: Jossey-Bass.
- [2] D. B. Fridsma, J. Thomsen (1998), *Representing medical protocols for organizational simulation: An information processing approach*, Computational and Mathematical Organization Theory, 4(1), 71-95.
- [3] R. M. Burton, and B. Obel (2004), *Strategic Organizational Diagnosis and Design: the Dynamics of Fit*, Kluwer Academic Publishers.
- [4] S. P. Robbins (1983), *Organization Theory: The structure and design of organizations*, Englewood Cliffs, N.J. Prentice-Hall, Inc.
- [5] H. A. Simon (1982), *Models of Bounded Rationality*, The MIT press, Cambridge, Massachusetts, London, England.
- [6] G.M. Levchuk, Y.N. Levchuk, J. Luo, K.R. Pattipati and D.L. Kleinman (2002), *Normative Design of Organizations – Part I: Mission Planning*, IEEE Trans. on SMC: Part A – Systems and Humans, Vol. 32, No. 3, pp. 346-359, May 2002.
- [7] A. Pete, D. L. Kleinman, and K. R. Pattipati (1994), *Structural congruence of tasks and organizations*, Proceedings of the 1994 Symp. on Command and Control Research and Decision Aids, NPS, Monterey, CA, pp. 168-175, 1994.
- [8] M. J. Prietula, K. M. Carley, and L. Gasser (1998), *Simulating Organizations: Computational Models of Institutions and Groups*, AAAI Press.
- [9] R.W. Pew, and A.S. Mavor (1998), *Modeling Human & Organizational Behavior: Application to military Simulations*, Washington, DC: National Academy Press.

- [10] M.R. Carrell, D.F. Jennings, and C. Hevrin (1997), *Fundamentals of Organizational Behavior*, Upper Saddle River, N.J.; Prentice Hall.
- [11] W.G. Scott, T.R. Mitchell, and P. H. Birnbaum (1981), *Organization theory: A structural and behavioral analysis*, Homewood, IL.: Richard D. Irwin.
- [12] J. Pfeffer (1996), *Understanding Organizations: Concepts and Controversies*, Research Paper No. 1378, Stanford Graduate School of Business.
- [13] C. Jacobsen and R. Bronson (1997), *Computer Simulated Empirical Tests of Social Theory: Lessons from 15 Years's Experience*, in R. Conte, R. Hegselmann and P. Terna (Eds.) *Lecture Notes in Economics and Mathematical Systems: Simulating Social Phenomena*, Berlin, Springer-Verlag.
- [14] P. Anderson (1999), *Complexity Theory and Organization Science*, *Organization Science*, 10(3), 216-232.
- [15] Website: <http://www.imaginethatinc.com/>
- [16] Air Force Instruction 13-1 AOC, Volume 3, *space, missile, command and control, Operational procedures, aerospace operations center*.
- [17] G. Casella and R. L. Berger (2001), *Statistical Inference*, Duxbury Press, 2<sup>nd</sup> edition.

# An Agent-based Simulation Model for Organizational Analysis

Sui Ruan  
Swapna S. Gokhale  
Krishna R. Pattipati

Dept. of Electrical and Computer Engineering  
University of Connecticut  
Contact: [krishna@engr.uconn.edu](mailto:krishna@engr.uconn.edu) (860) 486-2890

June 20-22, 2006

- Introduction
- Organizational model
  - ▶ Organizational components
  - ▶ Interrelationships among organizational components
- Agent behaviors
  - ▶ Situational awareness
  - ▶ Task scheduling
  - ▶ Event handling
- Illustrative example
- Summary and future work

# Introduction

## ■ Motivation

- ▶ Simulations based on computational organization theory
  - provide insights into the degree of match between tasks and organizational structure
  - identify bottlenecks
  - improve the quality and efficiency of an organization
- ▶ Traditionally, organizational theory describes an organization at an aggregate level
  - provides general qualitative predictions about its overall behavior

## ■ Our contribution

- ▶ Define and implement a “micro” model of the structure and behavior of the components in an organization
  - work activities, groups of agents, and the organizational structure
- ▶ Model information, expertise and knowledge as dynamic resources
  - can be acquired, and shared among agents
- ▶ Implement the model by a discrete-event simulator, *Extend<sup>TM</sup>*



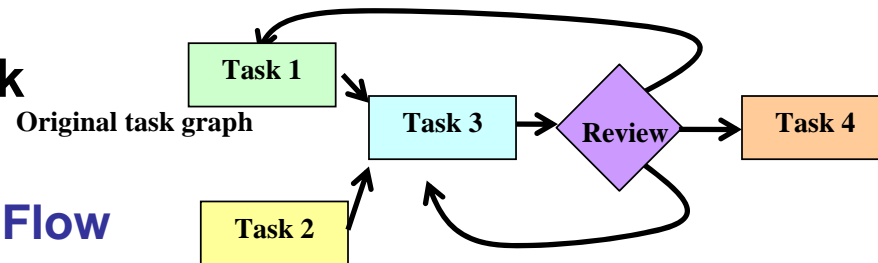
# Organizational Components - 1

## Resource

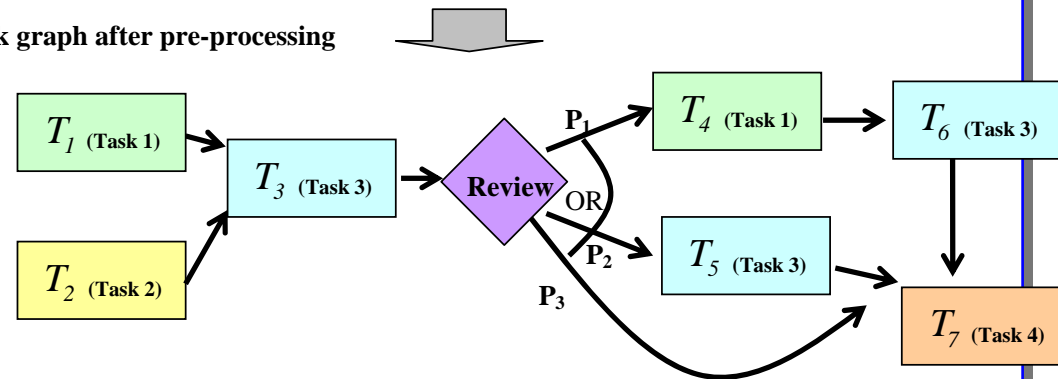
Identifier
Quantity
Transferability

## Work

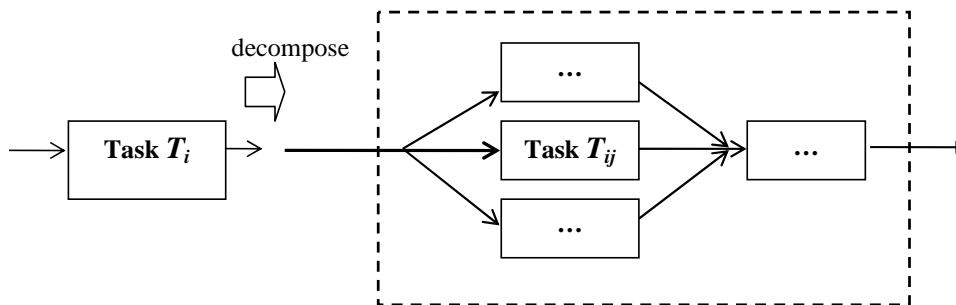
### a) Task Flow



Task graph after pre-processing



### b) Task-subtask decomposition

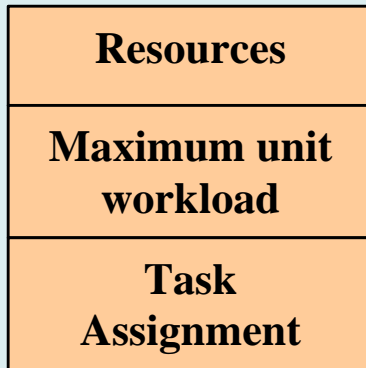


### c) Lowest level task characteristics

Required resources
Baseline processing time
Baseline unit workload
Resources to gain
Synchronization task set

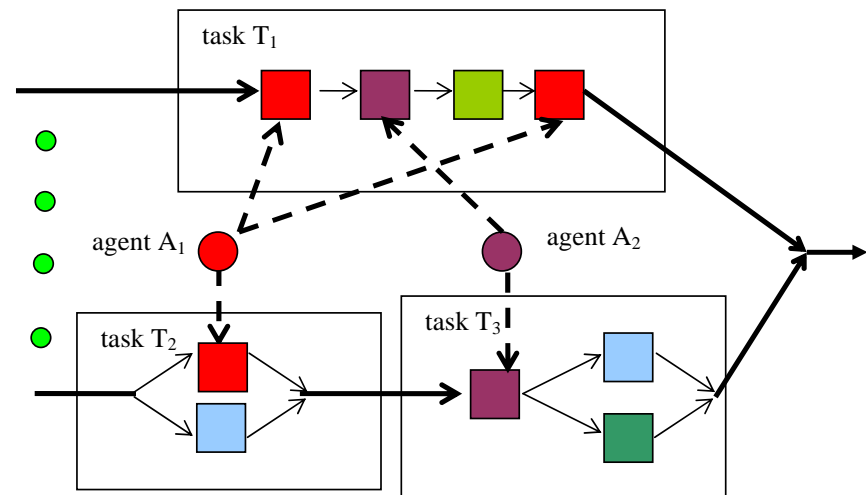
# Organizational Components - 2

## ■ Agent

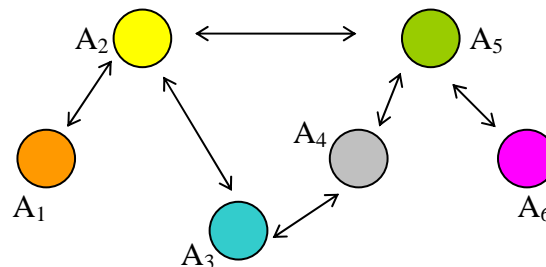


## ■ Organization

### ► Assignment of agents to tasks



### ► Communication channels among agents



# Interrelationships among organizational components

## ■ Modeling assumptions

- ▶ A relative degree of congruence exists between each pair of organizational components
- ▶ Basic hypothesis of the model is: *the greater the total degree of congruence among the various components, the more effective the organization will be*

## ■ Performance measures

- ▶ Mission completion time
- ▶ Mission quality and qualities of individual tasks
- ▶ Workload distribution among agents
- ▶ Communication and coordination load of individual agents and the organization

## ■ Situational Awareness

► At any time, an agent knows the status of all its own tasks:

– Status of a lowest-level task can be any one of the following:

- a) Not ready,
- b) Ready,
- c) Scheduled,
- d) Working,
- e) Completed

# Modeling Agent Behavior : Task Scheduling

## ■ Task Scheduling

For each ready task, the agent shall:

- 1) estimate its projected processing time, workload,
  - based on agent's resource capabilities and the task requirements
- 2) select task which has the earliest projected finish time and the projected workload is acceptable based on agent's current workload
- 3) communicate with other agent (s) who own the necessary transferable resources
- 4) if it is a task requiring synchronization
  - initiate synchronization message (s) with proposed start time
  - otherwise, start the task when all the resources are available

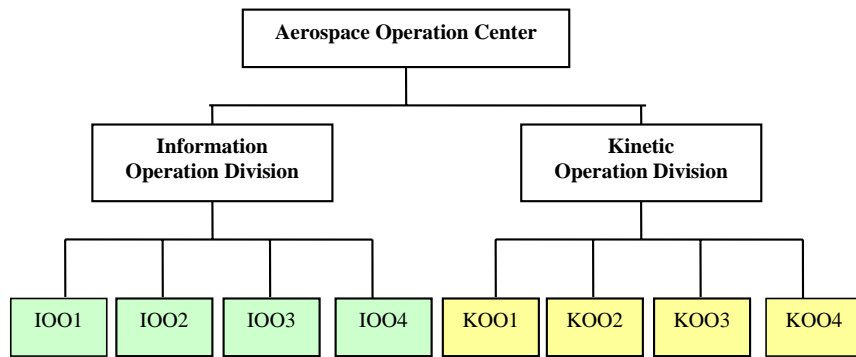
## ■ Event handling procedure of an agent

- ▶ when a task owned by an agent is completed, update:
  - task status
  - resource status
  - workload record
- ▶ when an agent receives transfer request for a resource
  - transfers the resource as requested
- ▶ when an agent receives a synchronization request
  - checks its availability at the requested time
  - if available, it acknowledges the synchronization message
  - else proposes a time which is the earliest available
  - sends a synchronization return message to the originator

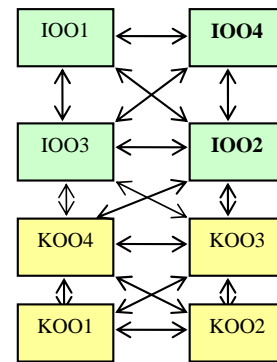
# Illustrative Example

## Air Operations Center (AOC)

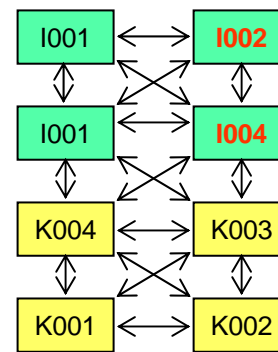
### Organization



a) Organization Chart



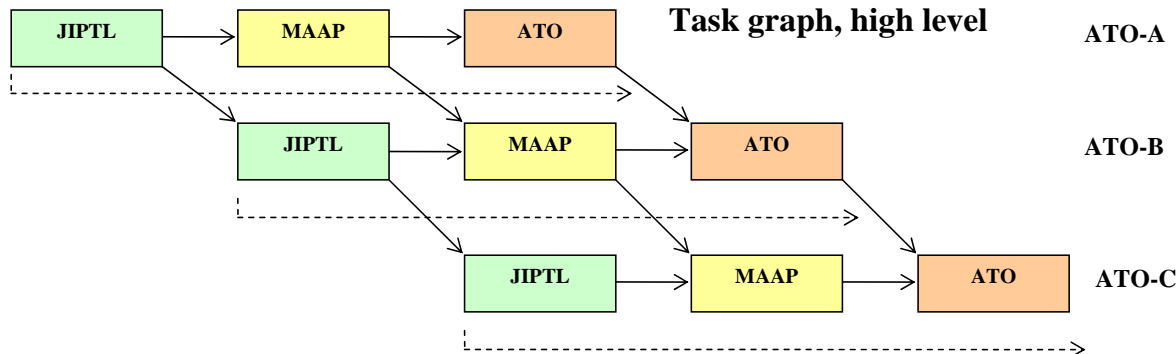
b) Communication Channels of ORG1



c) Communication Channels of ORG2

### Task Graph

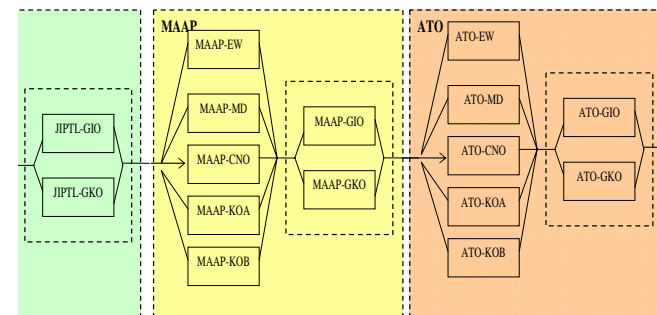
#### Task graph, high level



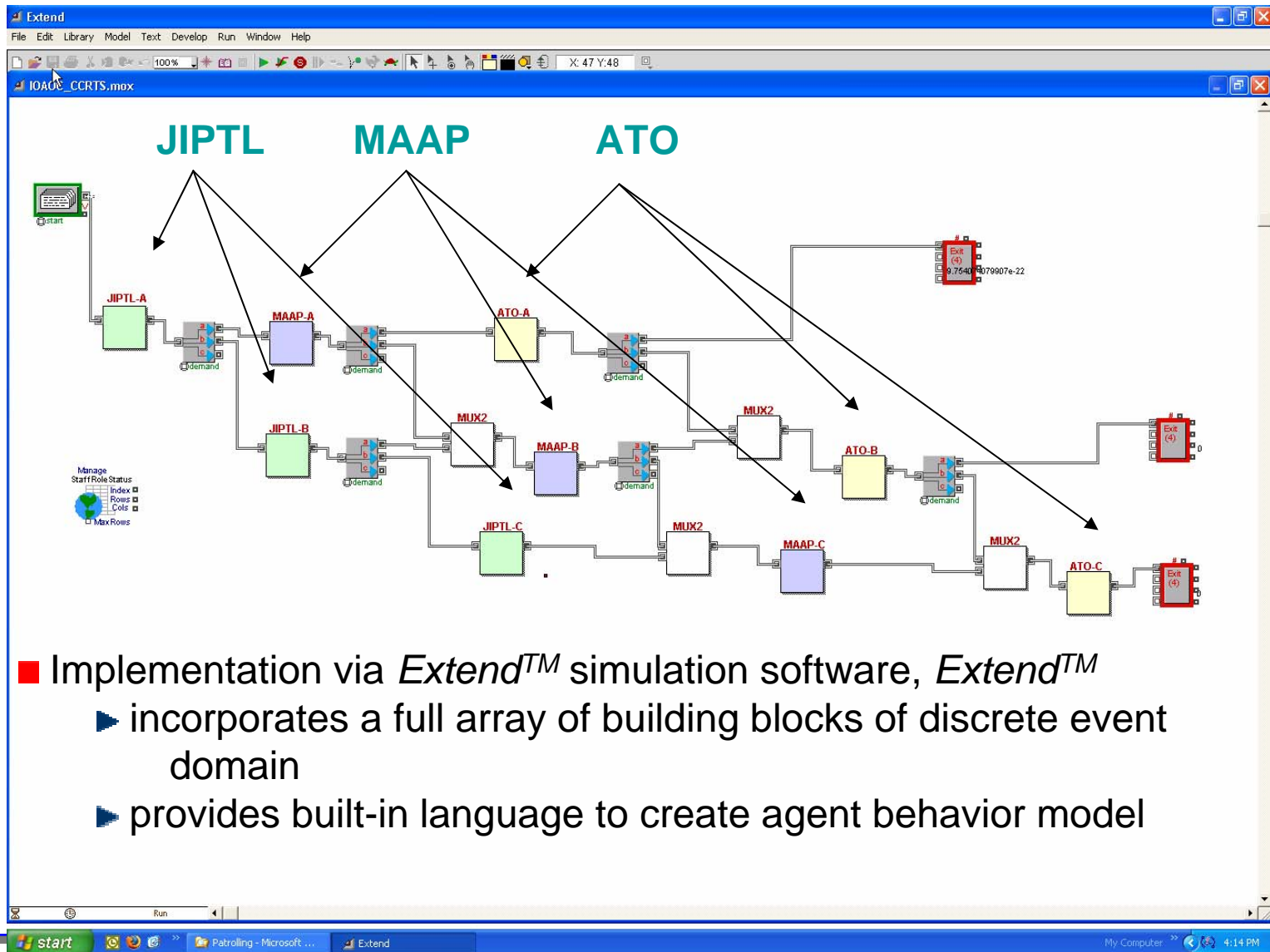
ATO-A

ATO-B

ATO-C



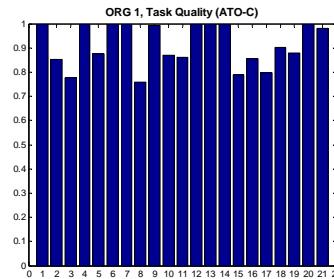
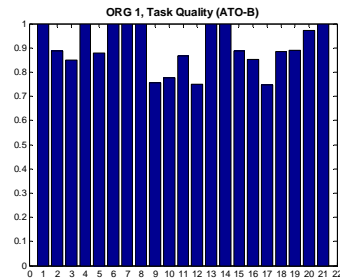
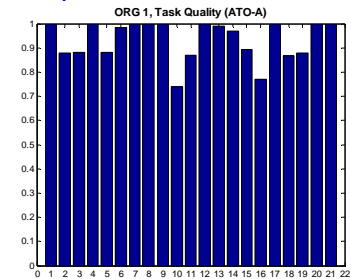
Task Graph, detailed



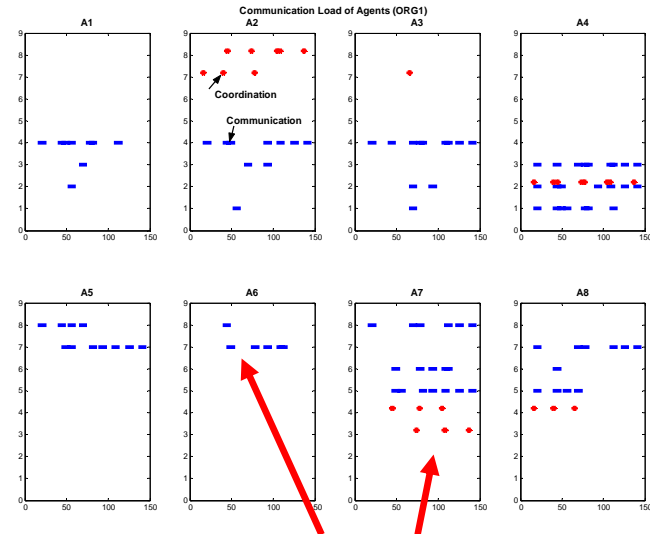
- Implementation via *Extend*<sup>TM</sup> simulation software, *Extend*<sup>TM</sup>
  - ▶ incorporates a full array of building blocks of discrete event domain
  - ▶ provides built-in language to create agent behavior model



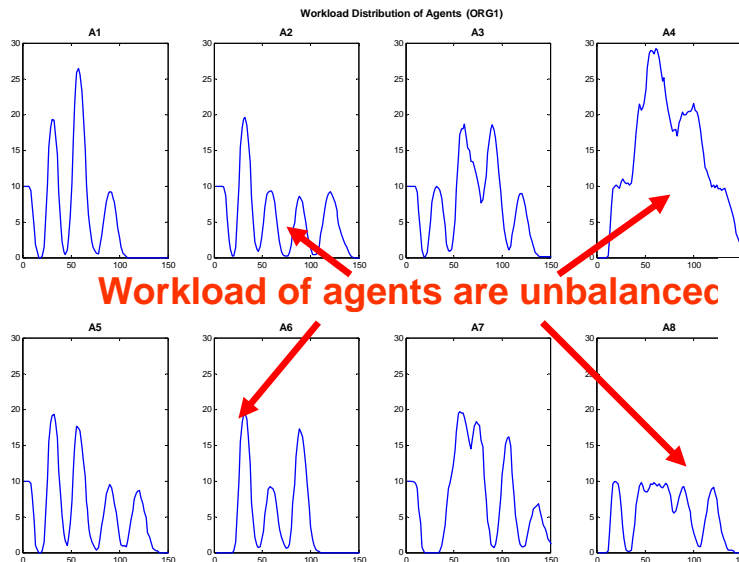
## ■ Performance Of Organization I



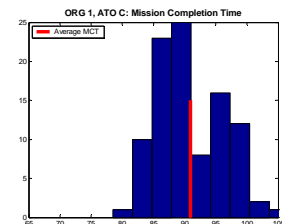
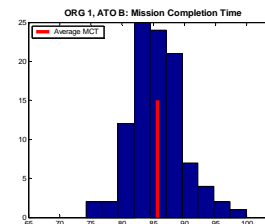
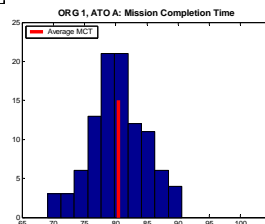
Some tasks are of low qualities



Communication load of agents are unbalanced



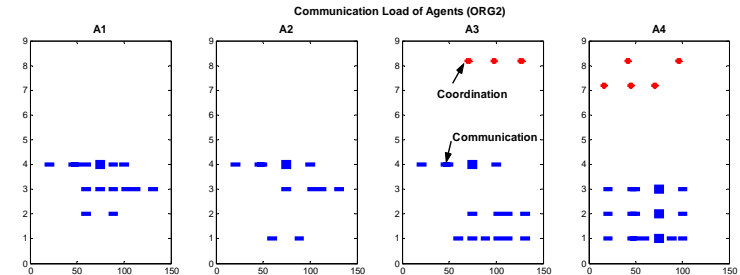
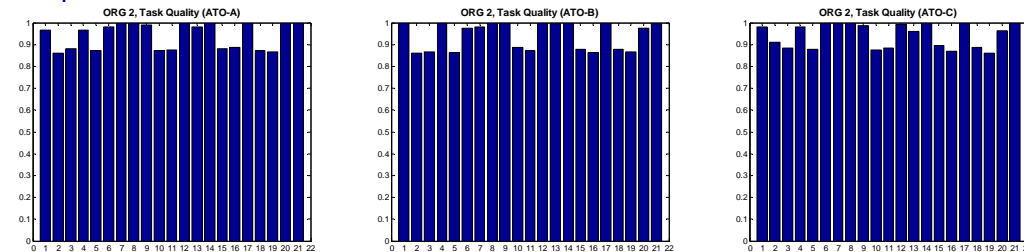
Workload of agents are unbalanced



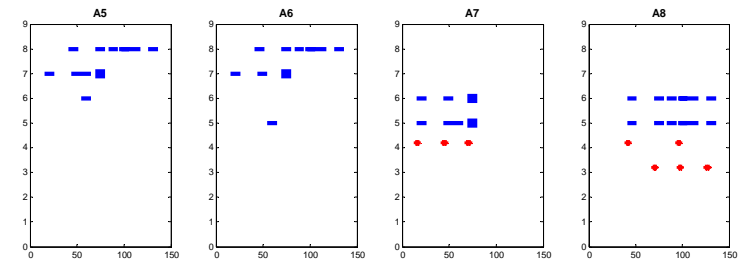
Mission completion time of each ATO is long

# Illustrative Example (4) Organizational Change

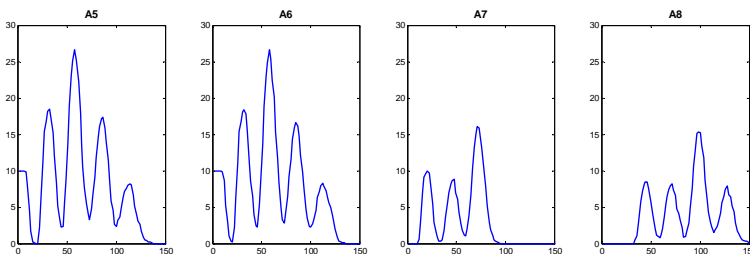
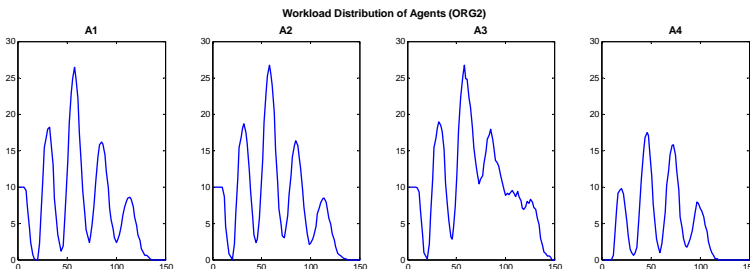
## ■ Performance Of Organization II



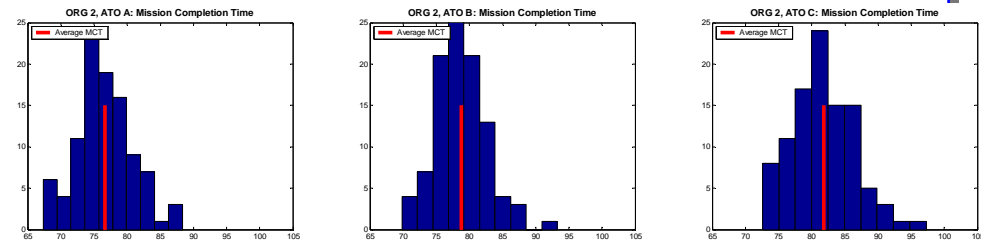
higher average task quality



smaller average communication load,  
and smaller communication load variance



smaller average agent workload,  
and smaller workload variance



shorter average mission completion time

## ■ In summary, we

- ▶ proposed a modeling and simulation methodology for organizations involved in project-based missions
  - modeled information, expertise and knowledge as dynamic resources, which agents can acquire and share
  - modeled interacting components, hierarchical arrangement of agents
  - implemented the organizational model using *Extend<sup>TM</sup>* simulation package, a discrete event simulator
- ▶ presented an example of an Air Operation Center (AOC)
  - to illustrate the potential of our modeling and simulation approach

## ■ In future work, we would consider

- ▶ more realistic and full-range of task interrelationships
- ▶ sophisticated agent behavior model
- ▶ impact of agent behavior on the task completion
- ▶ Introducing information propagation error (ambiguity)

# Questions?